Series de Tiempo

Modelos

Hernandez Torres Oliver
Mori Navarro Fredy
Purihuaman Manayay Agustin
Tejada Bances Yanaly
Velasquez Camacho Gabriela

29 de octubre de 2020

Análisis con pruebas Estadísticas

(«Raíz Unitaria (Dickey-Fuller) y estacionalidad en Excel | XLSTAT Support Center», s.f.)

Prueba Dickey fuller

$$H_0 : \textit{La serie tiene una raíz unitaria } (\textbf{\textit{no estacionaria}}) \quad (1)$$
$$H_1 : \textit{La serie no tiene una raíz unitaria}$$

Prueba Phillips-Perron (No paramétrica)

$$H_0 : \textit{Hay al menos una raíz unitaria } (\boldsymbol{\rho > 0.05}) \quad (2)$$
$$H_1 : \textit{No hay raíz unitaria (estacionaria)}$$

Prueba KPSS

$$H_0 : \textit{La serie es estacionaria} \quad (3)$$
$$H_1 : \textit{La serie no es estacionaria}$$

Modelo Autorregresivo

¿Que es el modelo **AR**?

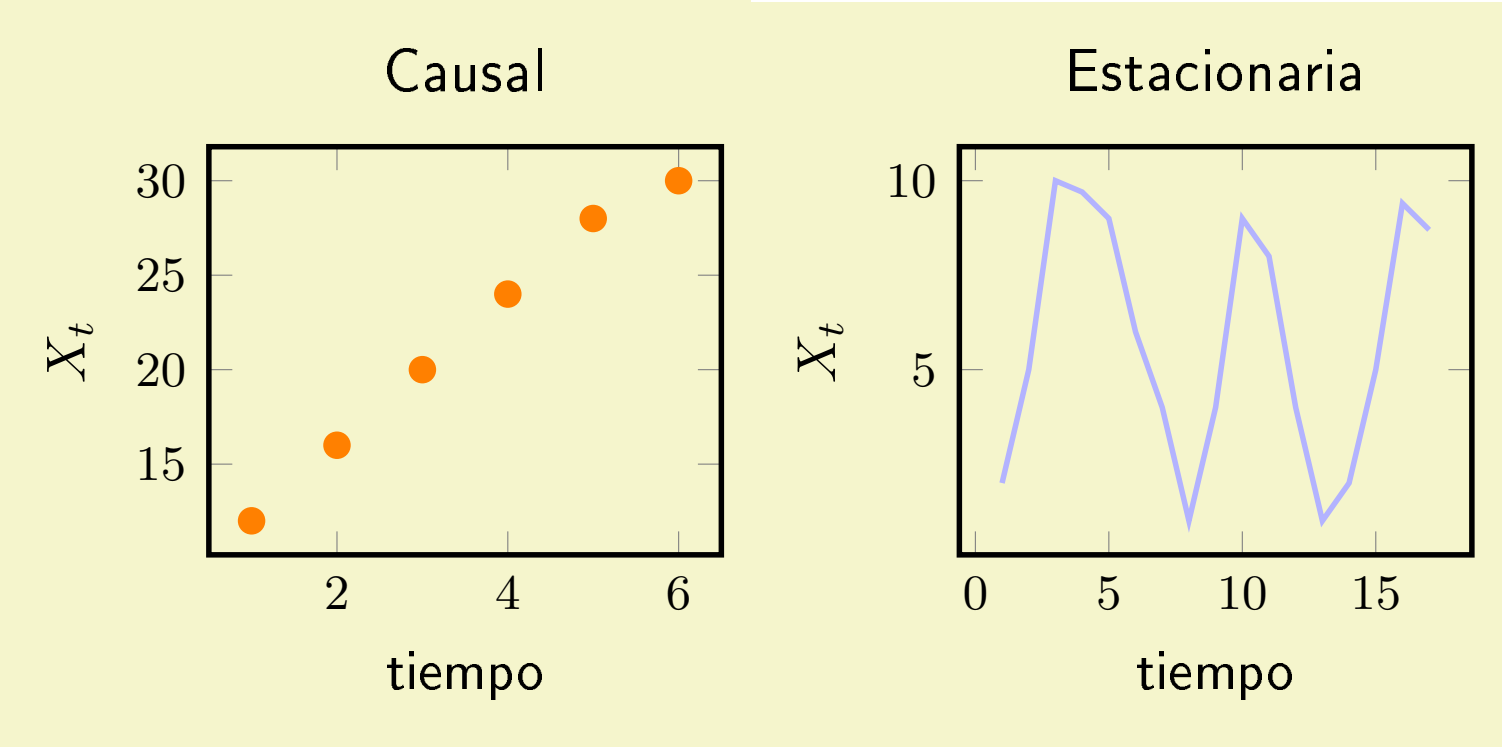
Es un modelo lineal que puede predecir algo basado en valores pasados de la misma cosa, se puede predecir basado en un cierto patrón. Se puede explicar como una combinación lineal de p-valores pasados, $X_{t-1}, \cdots, X_{t-p}$, agregado el error aleatorio de la misma serie (Tabassum, 2019).

$$X_t = c + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \epsilon_t$$

$\varphi_1, \cdots, \varphi_p$	Son parámetros en el modelo, *no puede ser cero*
X_t	valor esperado, es estacionario
x_t	Son valores pasados de la serie
ϵ_t	Es el ruido blanco ($\mu = 0 \quad \sigma^2 = 1$)
c	es la constante o deriva (drift)

Proceso Estacionario y Causal

El proceso estacionario significa que las propiedades estadísticas de un proceso que genera una serie de tiempo no cambian con el tiempo.**¿porque es importante?**
Por la razón de que los procesos estacionarios son mas fáciles de analizar. Un proceso causal significa que puede ser representado como un proceso lineal (Castagno, 2020).



Tipos AR(p)

Nota

AR(p) es un modelo auto regresivo de orden $\boldsymbol{p}$

Un modelo AR(1)

$\varphi_1 = 0$	X_t es equivalente a un ruido blanco
$\varphi_1 = 1$ y $c = 0$	X_t es equivalente a una caminata aleatoria
$\varphi_1 = 1$ y $c \neq 0$	X_t es equivalente a una caminata aleatoria con deriva (drift)
$\varphi_1 < 0$	x_t tiende a oscilar alrededor de la media

AR(1) modelo: $-1 < \varphi_1 < 1$

AR(2) modelo: $-1 < \varphi_2 < 1, \varphi_1 + \varphi_2 < 1, \varphi_2 - \varphi_1 < 1$

Cuando $p \geq 3$, las restricciones son mucho mas complejas

referencia tomada de (Hyndman y Athanasopoulos, 2018).

AR(1)

$$X_t = c + \varphi_1 x_{t-1} + \epsilon$$

AR(2)

$$X_t = c + \varphi_1 x_{t-1} + \boldsymbol{\varphi_2 x_{t-2}} + \epsilon$$

Nota

A medida que se agrega mas retardos (lags), el modelo se hace mas complejo con mas coeficientes.

Ejemplo

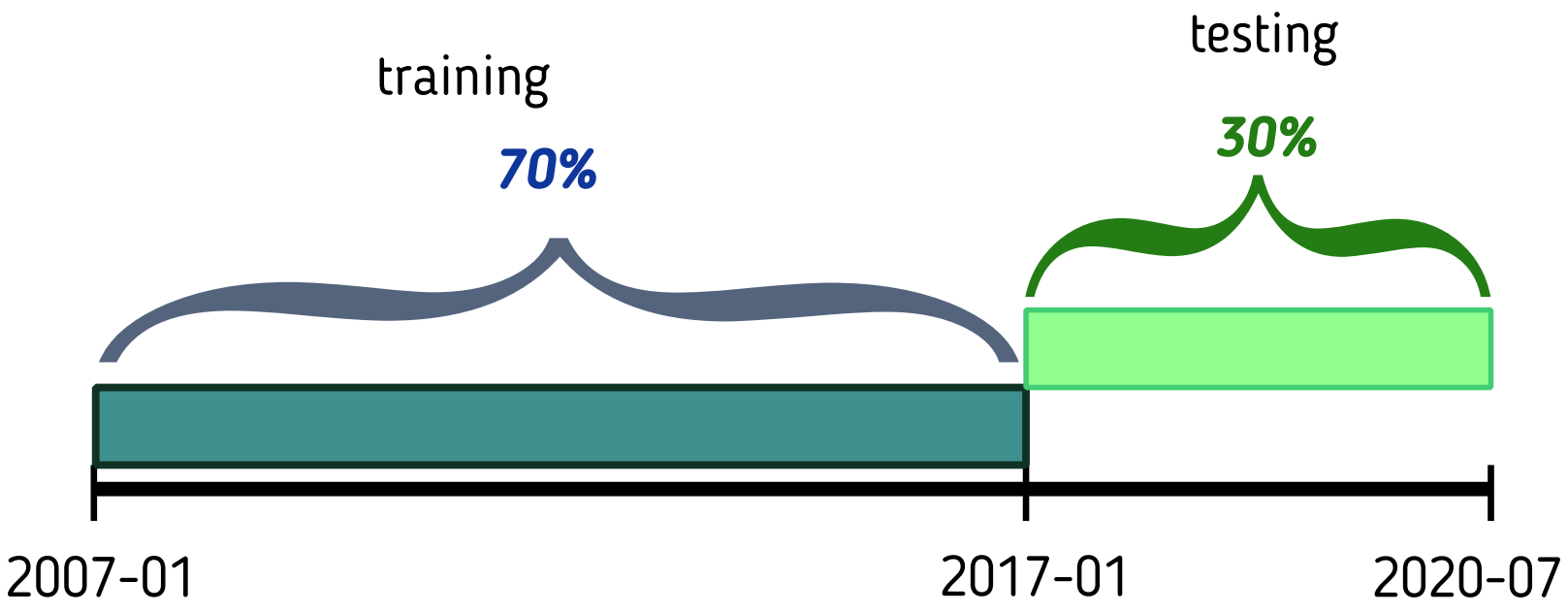
Usando la data de la Exportación del CACAO Peruano.

date	cacao
2007-01	2.13525
2007-02	2.466691
⋮	⋮
2020-07	14.74114

Se hizo el recorte respectivo desde:

train desde enero del 2007 hasta diciembre del 2016

test desde enero 2018 hasta julio del 2020



El recorte *train* nos permite entrenar el modelo o crearla, mientras que el *test* servirá para comparar la predicción usando el modelo creado por *train*.

Es importante que el pronostico tenga la misma cantidad de elementos que el *test*.

Usando Python

cargando librerias

```
import pandas as pd
import numpy as np
from pandas.plotting import lag_plot
from statsmodels.graphics.tsaplots import plot_acf,
↪  plot_pacf
from statsmodels.tsa.stattools import adfuller
from matplotlib import pyplot
from statsmodels.tsa.arima_model import ARMA
from pandas.plotting import lag_plot,
↪  autocorrelation_plot,hist_series
import warnings
warnings.filterwarnings("ignore")
from sklearn.metrics import mean_squared_error,
↪  mean_absolute_error, r2_score
```

(Pedregosa y col., 2011)

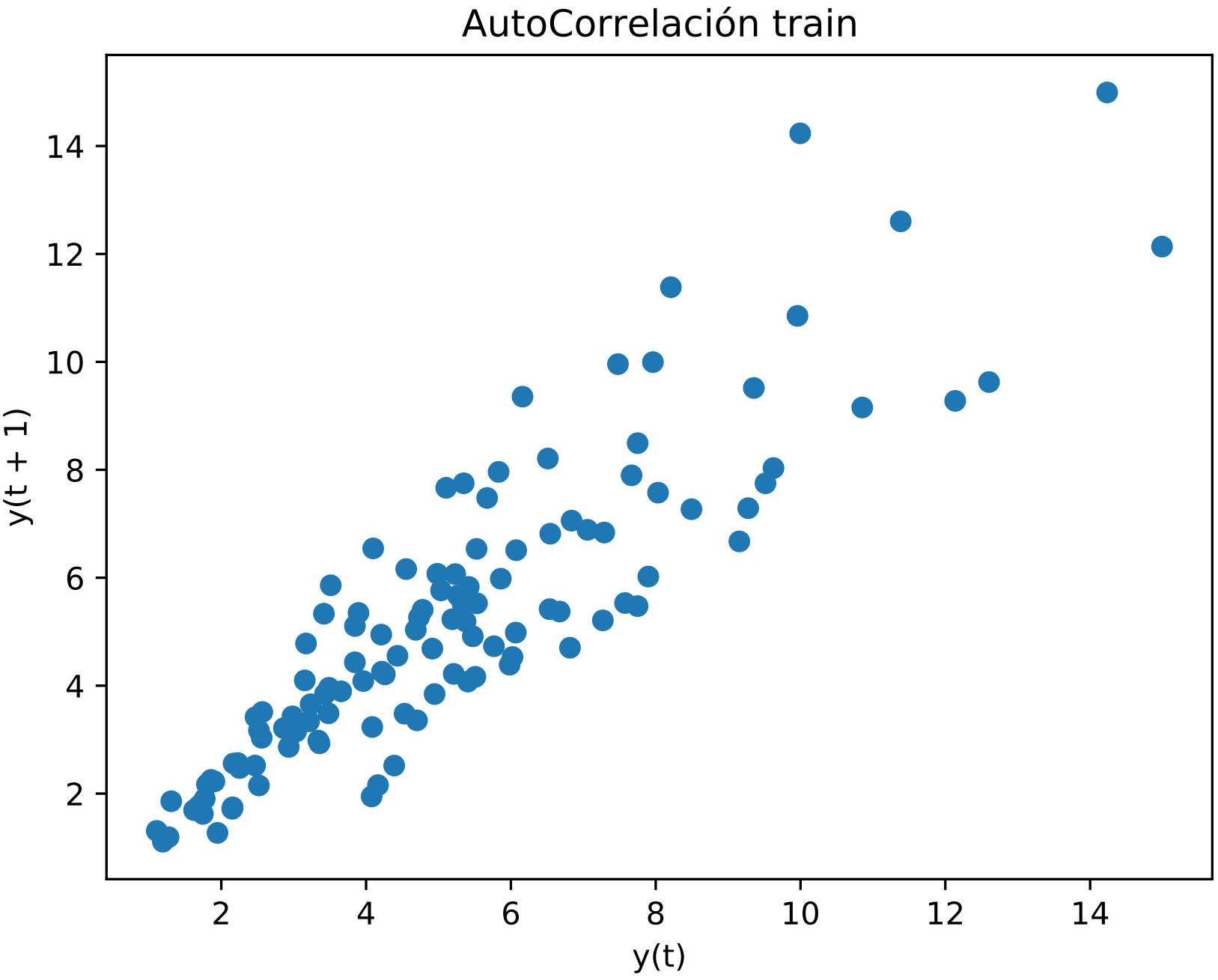
Importando y recortando

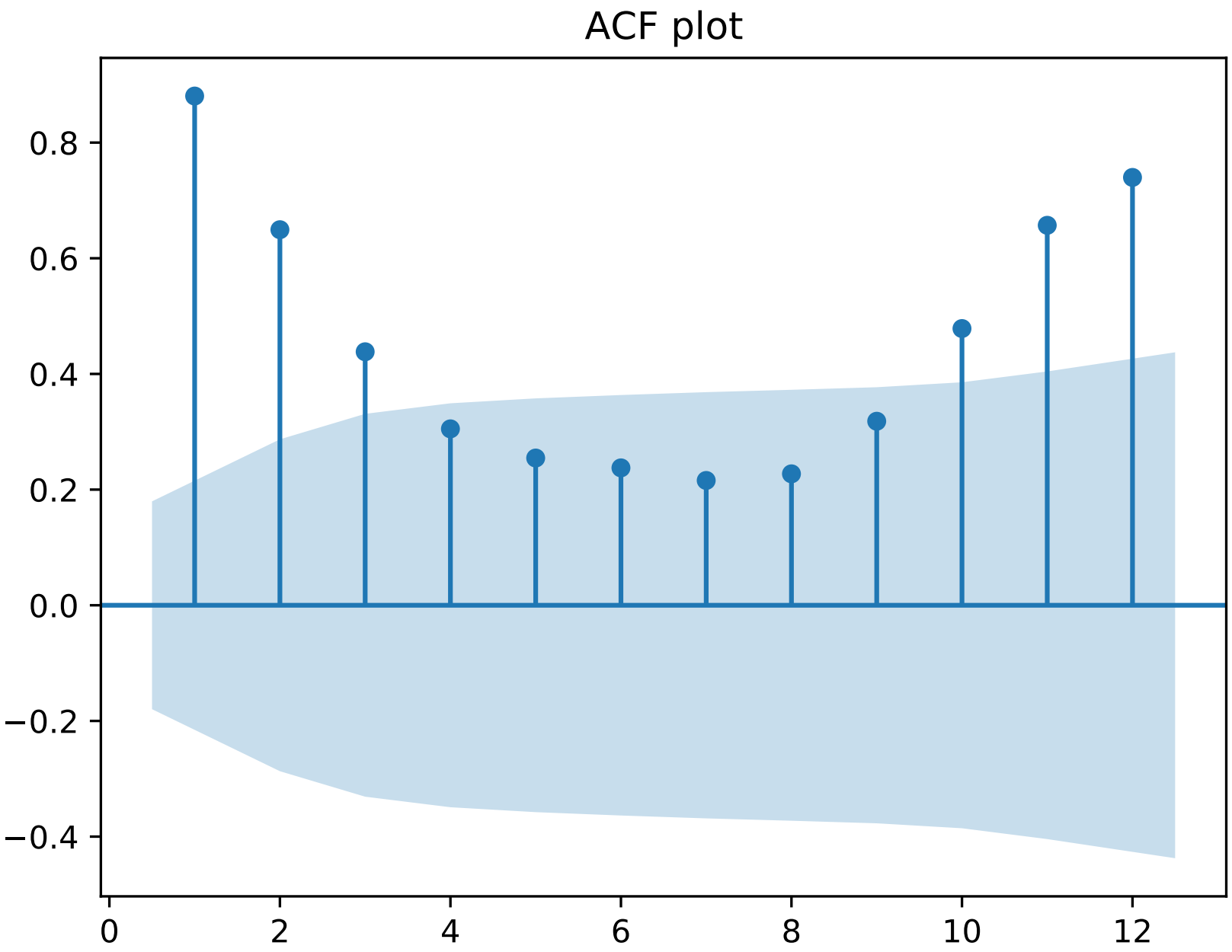
```
dt = pd.read_csv("cacaos.csv", header=0,
↪  parse_dates=[0], index_col=0, sep=",")
train, test = dt.iloc[1:120], dt.iloc[121:]
```

¿Que pasaria si omito la estacionariedad en los datos?

```
lag_plot(train)
pyplot.title("AutoCorrelación train")
plot_acf(train,zero=False, lags=12)
pyplot.title("ACF plot")
pyplot.show()

```


AutoCorrelación train
y(t + 1)
y(t)
2
4
6
8
10
12
14


ACF plot
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
2
4
6
8
10
12

Aplicando modelo para estimación

```
# modelo
model = ARMA(train, order=(9,0))
fit = model.fit()
```

se ha usado un modelo AR(9) porque es el que mejor se aproxima a la realidad

```
print(fit.summary())
```

Generando un resumen

```
==============================================================
         coef    std err     z       P>|z|    [0.025  0.975]
--------------------------------------------------------------
const    5.6020   1.958    2.860   0.004     1.764   9.440
ar.L1    1.3952   0.082   16.972   0.000     1.234   1.556
ar.L2   -0.7594   0.150   -5.060   0.000    -1.054  -0.465
ar.L3    0.2601   0.166    1.570   0.116    -0.065   0.585
ar.L4   -0.3021   0.167   -1.812   0.070    -0.629   0.025
ar.L5    0.3150   0.167    1.887   0.059    -0.012   0.642
ar.L6    0.0046   0.166    0.028   0.978    -0.321   0.330
ar.L7   -0.0958   0.164   -0.582   0.560    -0.418   0.227
ar.L8   -0.2745   0.152   -1.803   0.071    -0.573   0.024
ar.L9    0.4312   0.086    5.017   0.000     0.263   0.600
                          Roots
==============================================================
```

En la tabla anterior cuando $\rho < 0.05$, se dice que φ_i es significativamente diferente de 0.

Nota: ar.Li son los retrasos φ_i

Creando una estimación

```
fit = model.fit()
print(fit.summary())
print(fit.k_ar)
print(fit.params)
predic = fit.predict(start=len(train),
↪  end=len(train)+len(test)-1 )
```

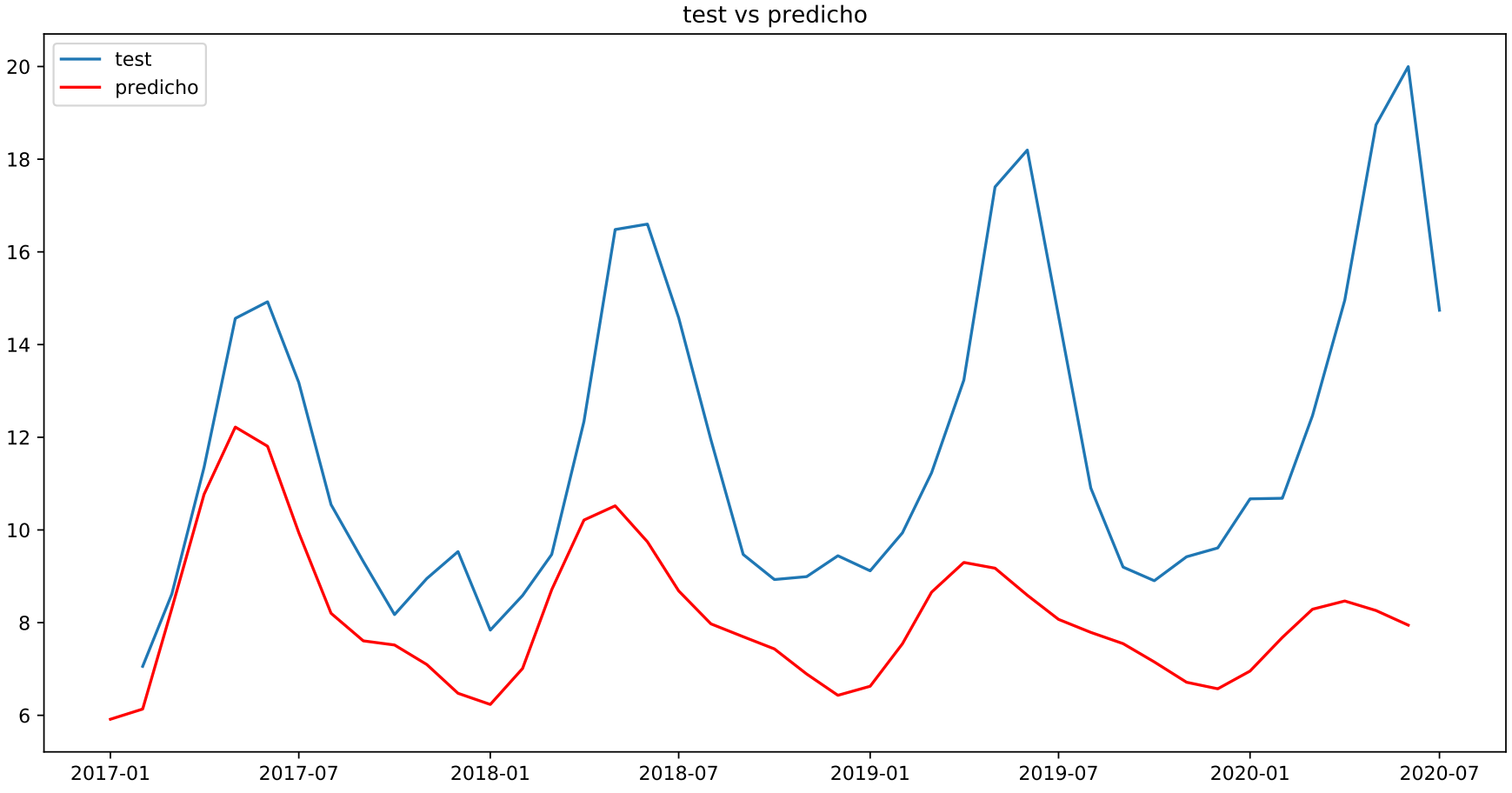
Función para analizar la fiabilidad

```
# ----- Fiabilidad del modelo -----
def timeseries_evaluation_metrics_func(y_true, y_pred):
    def mean_absolute_percentage_error(y_true, y_pred):
        y_true, y_pred = np.array(y_true), np.array(y_pred)
        return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
    print('Evaluation metric results:-')
    print(f'MSE is : {mean_squared_error(y_true,y_pred)}')
    print(f'MAE is : {mean_absolute_error(y_true,y_pred)}')
    print(f'RMSE is : {np.sqrt(mean_squared_error(y_true,
    ↪  y_pred))}')
    print(f'MAPE is :
    ↪  {mean_absolute_percentage_error(y_true,y_pred)}')
    print(f'R2 is : {r2_score(y_true, y_pred)}',end='\n\n')
```

Resúmenes de fiabilidad del modelo

AR(p)	MSE	MAE	RMSE	MAPE	R^2
AR(1)	51.656	6.327	7.187	50.352	-3.774
AR(2)	54.220	6.556	7.363	52.431	-4.011
AR(3)	53.361	6.486	7.305	51.793	-3.931
AR(4)	53.228	6.474	7.296	51.685	-3.919
AR(5)	51.219	6.290	7.157	50.019	-3.733
AR(6)	53.345	6.479	7.304	51.735	-3.93
AR(7)	51.156	6.290	7.152	50.014	-3.728
AR(8)	35.461	4.938	5.955	38.226	-2.277
AR(9)	20.212	3.668	4.496	30.052	-0.868

```
print(timeseries_evaluation_metrics_func(y_true=
↪  test,y_pred=predic))
```


test vs predicho
test
predicho
20
18
16
14
12
10
8
6
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
2020-07

Algunas funciones R que resultan igual

```r
tb = read.csv(file.choose(),header = T)[,2]
yt = ts(data = tb,start = c(2007,1),end = c(2020,7),frequency = 12)
# recortar data
train = window(yt,start=c(2007,1), end=c(2016,12))
test = window(yt, start = c(2017,1))
# acf(x = yt,lag.max = 4)
lag.plot(x = yt,lags = 12)
acf(x = train)
pacf(x = train)
fit = (arima(x = train,order = c(9,0,0)))
predicho = predict(fit,n.ahead = 43 )$pred
plot.ts(test, lwd= 2)
lines(predicho, col="red", lwd=2)
```

Corrigiendo el modelo usando transformación y diferencia

```python
# ---- aplicando diferenciacion -------
tb = np.diff(np.log(dt.cacao), n=2)
train2, test2 = tb[2:118], tb[118:]
```

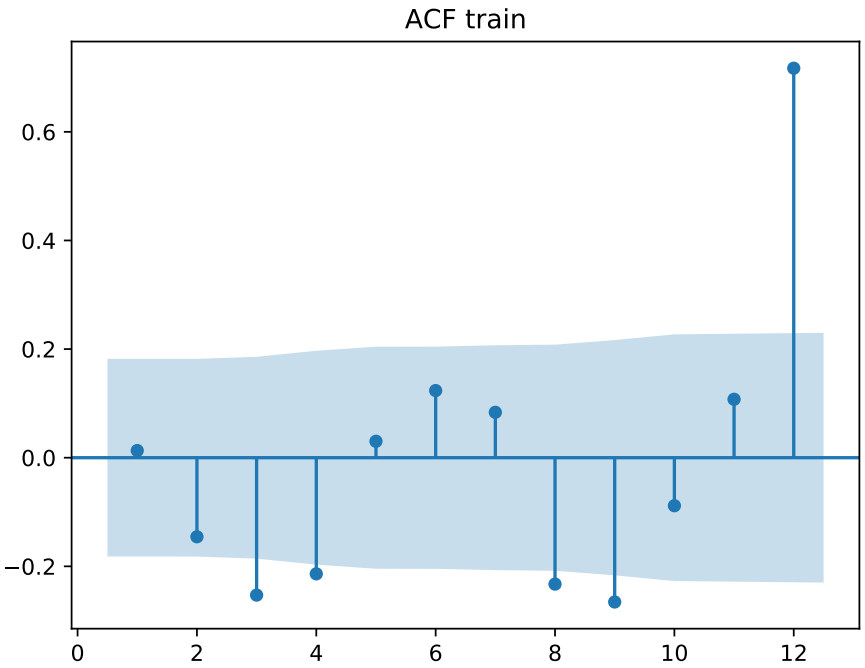
Comprobando con la prueba de Dickey Fuller

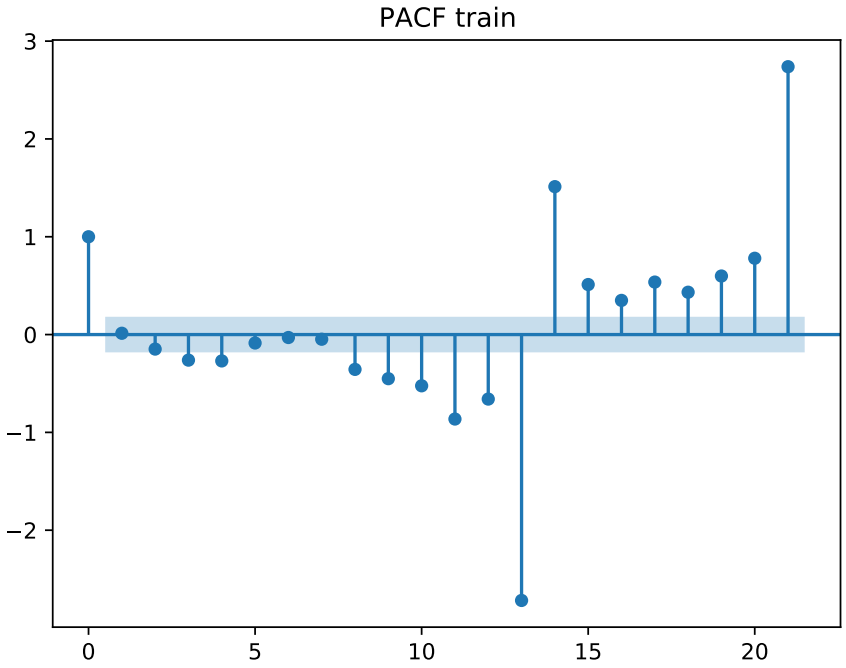
```python
adf = adfuller(train2)
print(f'ADF Estadistico: {adf[0]}')
print(f'ADF p-valor:  {adf[1]}')
```

Estadístico	p-valor
-8.6788	4.342257047351299e-14

Siendo ($p < 0.05$), la serie es Estacionaria

Graficos ACF y PACF





Casos en un AR()

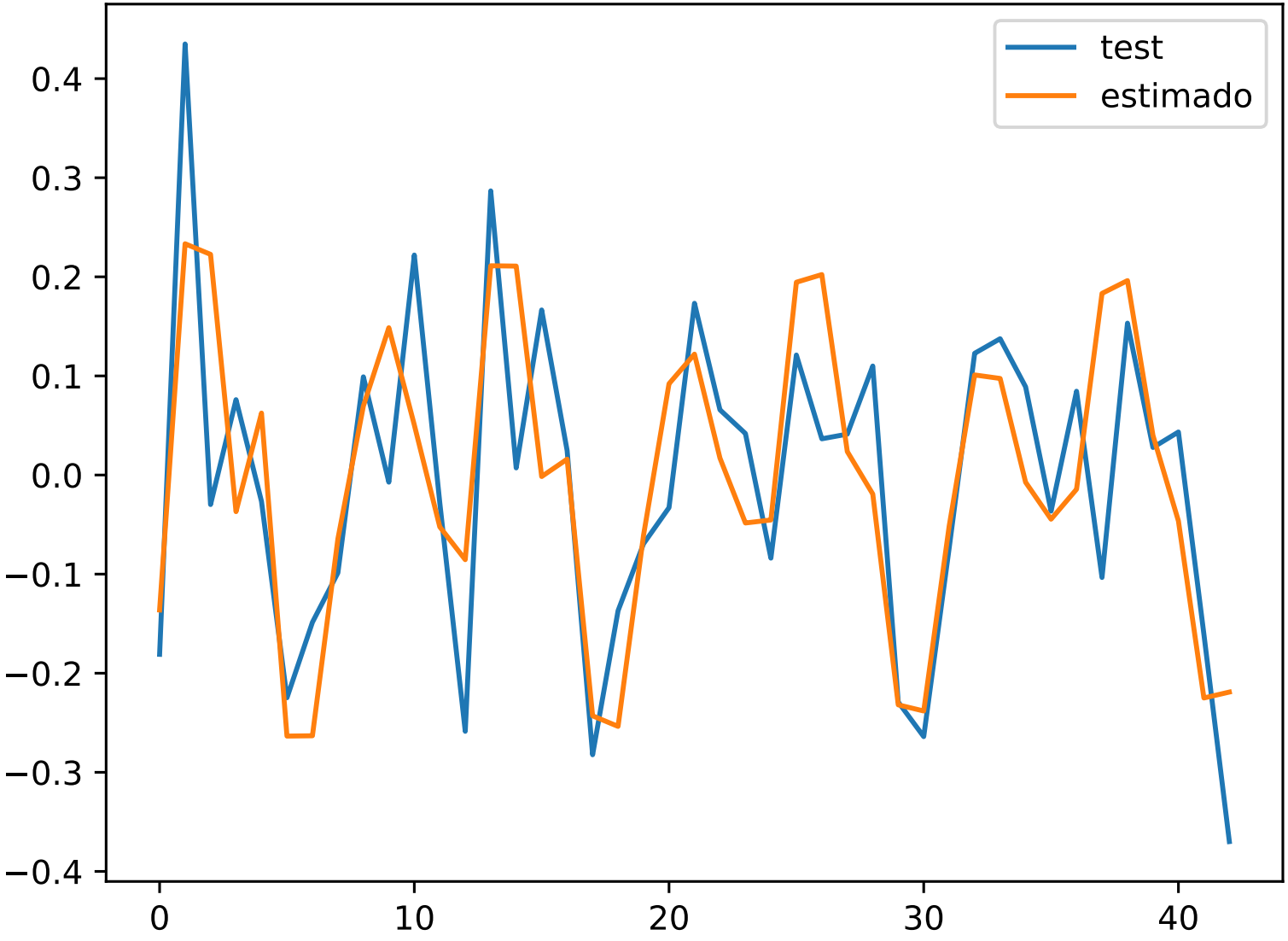
PACF representa retrasos significativos

ACF debe ser geométrico.

Analizando Fiabilidad

AR(p)	MSE	MAE	RMSE	MAPE	R^2
AR(1)	0.0256	0.126	0.160	100.12	-0.001
AR(2)	0.025	0.125	0.159	100.08	0.009
AR(3)	0.0254	0.125	0.159	100.722	0.007
AR(4)	0.026	0.125	0.162	104.53	-0.024
AR(5)	0.0267	0.126	0.163	105.156	-0.04
AR(6)	0.0268	0.126	0.164	104.58	-0.044
AR(7)	0.027	0.126	0.163	102.095	-0.039
AR(8)	0.024	0.122	0.154	105.37	0.078
AR(10)	0.017	0.103	0.13	182.963	0.3413
AR(11)	**0.0126**	0.088	0.112	219.91	**0.5096**
AR(12)	0.013	0.089	0.114	226.41	0.489

Nota: Es aplicando transformación


test
estimado
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0
10
20
30
40

Promedio Móvil MA()

Promedio Móvil

¿Que es MA(q)?

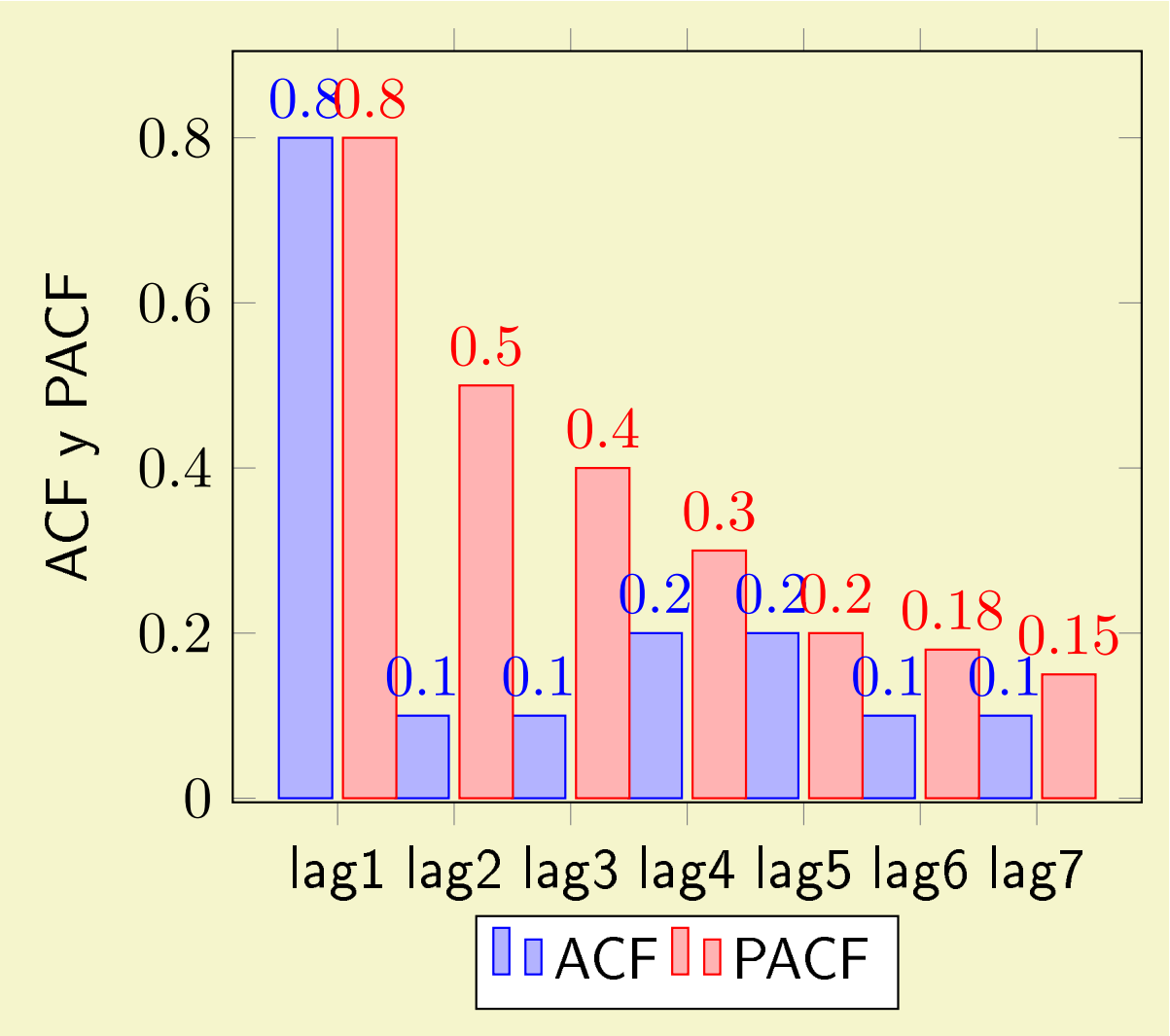
Es un modelo que utiliza la dependencia entre una observación y un error residual de un modelo de promedio móvil aplicado a observaciones retrasadas (lags), q es un parámetro de cuántas observaciones retrasadas se deben tomar. Contrario al modelo AR(p) , el modelo finito es siempre estacionario (Kang, 2017).

$$X_t = c + \theta\epsilon_{t-1} + \epsilon_t$$

Leyendo un ACF y PACF

	ACF	PACF
AR	Geométrico	p retrasos significativos (orden)
MA	q retrasos significativos (orden)	Geométrico
ARMA	Geométrico	Geométrico

(«READING ACF AND PACF PLOTS», s.f.)



En esta gráfica, el ACF es significativo solo una vez (en realidad, la primera entrada en el ACF siempre es significativo, ya que no hay retrasos en la primera entrada - es la correlación consigo mismo), mientras que el PACF es geométrico. Por tanto, es un proceso MA (1).

Modelos Mixtos ARMA

ARMA(p,q)

(p) valores retardos

(q) retardos del error

Modelo ARIMA

AR	Auto Regresión
I	Integrado (Diferenciación de observaciones,permitir estacionariedad)
MA	Promedio Móvil (observación y error residual)

p	Numero de observaciones de retardo en el modelo, *orden de retraso (lag)*
d	Numero de veces que se diferencian las observaciones en bruto (raw), *grado de diferenciación*
q	Orden de la media móvil

Bibliografía

Castagno, P. (2020). *Autoregressive Model (AR)*. Consultado el 15 de octubre de 2020, desde https://medium.com/@patriziacastagnod/autoregressive-model-ar-83a19a48ec92

Hyndman, R. y Athanasopoulos, G. (2018). *Forecasting: principles and practice* (2.ª ed.). OTexts.

Kang, E. (2017). Time Series: ARIMA Model. Consultado el 16 de octubre de 2020, desde https://medium.com/@kangeugine/time-series-arima-model-11140bc08c6

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

Raíz Unitaria (Dickey-Fuller) y estacionalidad en Excel | XLSTAT Support Center. (s.f.). Consultado el 25 de octubre de 2020, desde https://help.xlstat.com/s/article/raiz-unitaria-dickey-fuller-y-estacionalidad-en-excel?language=es

READING ACF AND PACF PLOTS. (s.f.). Consultado el 16 de octubre de 2020, desde https://rinterested.github.io/statistics/arima.html

Tabassum, N. (2019). *Time series: Autoregressive Model*. Consultado el 15 de octubre de 2020, desde https://medium.com/@nazia.tabassum_48787/time-series-autoregressive-model-9703746fb74b